

| 质量属性分类 | | | 序号 | 描述 | 得分 |
|----------------|------|-------|----|--|----|
| 软件质量 (外部质量) | 功能 | | 1 | 对原roomuserinfo进行了封装，使得上层view使用相关属性更加方便，同时不用关注过多其他不必要的 | 30 |
| | | | 2 | | |
| | | | 3 | | |
| | | | 4 | | |
| | | | | | |
| | 性能 | | 1 | 在用户变化时，增加了handler队列机制，每次发送变化时先移除上一次，在大量用户登录的情况下能一定程度上缓解主线程压力 | 10 |
| | | | 2 | 使用单例模式对用户数据进行存放，在只有与UI展示相关属性变化时才通知UI更新，减少UI刷新次数 | |
| | | | | | |
| | | | | | |
| | 其他 | | | 采用MVP设计，很好的分离了数据与UI，将逻辑封装到present中，方便后续扩展和维护 | 10 |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| 代码质量 (内部质量) | 代码设计 | | 1 | 采用MVP设计，很好的分离了数据与UI，将逻辑封装到present中，方便后续扩展和维护 | 16 |
| | | | 2 | 采用了观察者模式，数据驱动UI更加便捷 | |
| | | | 3 | 重新包装了一个新接口空实现，方便实现者不用关注与自己无关的方法实现 | |
| | | | 4 | 观察者模式部分接口过多，在大量数据调用时可能会有性能问题，建议按职责对接口进一步细分 | |
| | | | | | |
| | | | | | |
| | 代码实现 | 编程规范 | 1 | 部分代码注释偏少 | 8 |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | 代码坏味道 | 1 | 方法单一性原则，数据初始化应放到定义好的初始化方法中 | 18 |
| | | | 2 | 平行继承体系，重新包装了接口做了空实现简化了其他类使用时需要关注的方法，但是会造成其他类继承过多不需要使用的方法 | |
| | | | | | |
| | | | | | |
| | | | | | |
| 总分： | | | | | 92 |